



Causal Models in Requirement Specifications for Machine Learning: A vision

Downloaded from: <https://research.chalmers.se>, 2025-09-25 23:23 UTC

Citation for the original published paper (version of record):

Heyn, H., Mao, Y., Weiß, R. et al (2025). Causal Models in Requirement Specifications for Machine Learning: A vision. Proceedings of the ACM SIGSOFT Symposium on the Foundations of Software Engineering: 1402-1405. <http://dx.doi.org/10.1145/3696630.3731614>

N.B. When citing this work, cite the original published paper.



Causal Models in Requirement Specifications for Machine Learning: A vision

Hans-Martin Heyn
hans-martin.heyne@gu.se
Chalmers University of Technology
and University of Gothenburg
Göteborg, Sweden

Yufei Mao
Roland Weiß
{yufei.mao,rolandweiss}@siemens.com
Siemens AG
München, Germany

Eric Knauss
Eric.Knauss@cse.gu.se
Chalmers University of Technology
and University of Gothenburg
Göteborg, Sweden

Abstract

Specifying data requirements for machine learning (ML) software systems remains a challenge in requirements engineering (RE). This vision paper explores causal modelling as an RE activity that allows the systematic integration of prior domain knowledge into the design of ML software systems. We propose a workflow to elicit low-level model and data requirements from high-level prior knowledge using causal models. The approach is demonstrated on an industrial fault detection system. This paper outlines future research needed to establish causal modelling as an RE practice.

CCS Concepts

• **Software and its engineering** → **Requirements analysis**;
Reusability; *Software prototyping*.

Keywords

AI Engineering, Causal Modelling, Data Requirements, Requirements Engineering

ACM Reference Format:

Hans-Martin Heyn, Yufei Mao, Roland Weiß, and Eric Knauss. 2025. Causal Models in Requirement Specifications for Machine Learning: A vision. In *33rd ACM International Conference on the Foundations of Software Engineering (FSE Companion '25)*, June 23–28, 2025, Trondheim, Norway. ACM, New York, NY, USA, 4 pages. <https://doi.org/10.1145/3696630.3731614>

1 Introduction

Rahimi et al. called for more attention towards the ability of specifying software with machine learning (ML) components [24]. Many industrial applications require *robustness* of ML models against changes in input data distribution [4]. A key reason for lacking robustness is the difficulty of specifying ML models, because “if input and/or output data are high-dimensional, both defining preconditions and detailed function specifications are difficult” [18]. Robustness against context changes can only be tested if the expected operational context is explicitly defined, for instance through contextual requirements [16, 17]. However, assumptions about the operational context are often implicit in the design process [21], such as in the selection of the training dataset. Recent surveys on requirements engineering (RE) confirm that specifying training

data for ML models remains an open challenge [1, 10, 23]. Current RE techniques struggle to translate high-level functional and non-functional requirements into data requirements [2, 26]. This leads to an *underspecification* causing variability in implementation choices and a lack of robustness against context changes [6].

A possible way to address underspecification is reasoning about expected causal relationships in the ML system’s operational context. Typically, ML cannot infer causality from data alone [22]. An ML model learns a probabilistic representation from data that seems compatible in a training context, but its performance might deviate drastically in a different operational context as statistical correlations do not capture true causal mechanisms [5]. Addressing this challenge requires incorporating prior domain knowledge and causal reasoning into the design of ML systems.

This vision paper proposes causal modelling to communicate *prior knowledge* about causal relations in the operational context. We argue that by formulating prior domain knowledge as causal models we can derive requirements towards data, as well as deduce rules for runtime verification. This will lead to causally motivated requirements specifications for software with ML.

Objective of this vision paper. First, we outline our vision of integrating causal modelling as an RE activity for ML systems. Then, we illustrate its application in eliciting data requirements for an industrial prototype of an ML-based cooling fault-detection system for electric motors. Finally, we discuss a research agenda to explore the potential of causal modelling as an RE activity for ML systems.

2 Related Work

The potential of using causal modelling as part of RE activities is not yet fully explored [12]. Fischbach et al. proposed an NLP-based process to extract and structure causal relationships from natural language [7, 9]. A tree recursive neural network (TRNN) model was trained to detect causality in natural language requirements using logical markers such as conjunctions and negations [15]. They further developed an approach to convert extracted causal relationships into a DAG-like structure to automatically generate test cases [8]. Maier et al. proposed modelling cause-effect relationships as part of scenario-based testing for automotive system safety [19]. Maier et al. also introduced the concept of “CausalOps”, an industrial lifecycle framework for causal models [20]. Gren et Brentsson Svensson proposed Bayesian Data Analysis (BDA) to evaluate the outcome of experiments on the effect of obsolete requirements on software effort estimation [14]. Similarly, Frattini et al. investigated the impact of requirements quality defects on domain modelling by using BDA and causal reasoning in a controlled experiment [11]. While the latter two studies do not use causal modelling



This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.

FSE Companion '25, Trondheim, Norway

© 2025 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-1276-0/2025/06

<https://doi.org/10.1145/3696630.3731614>

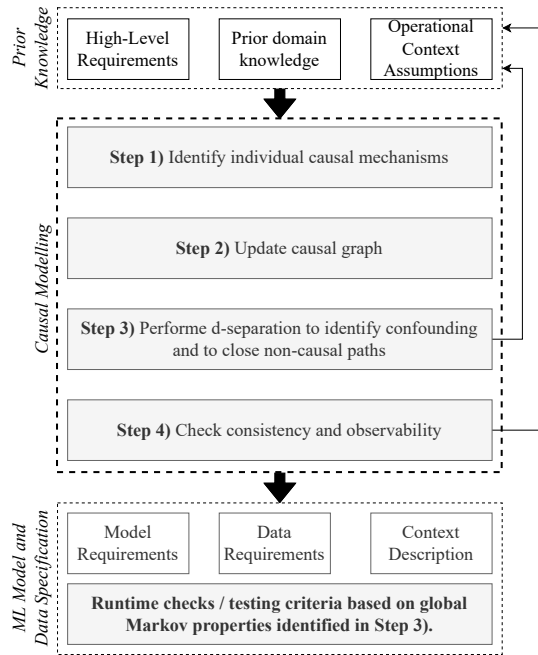


Figure 1: A proposed workflow for Causal RE

as an explicit RE activity, these studies demonstrates the potential of applying causal reasoning to RE activities.

3 Causal modelling as an RE activity

In a typical ML development pipeline, causal modelling would be a step between problem definition and data collection as it allows to formalise domain knowledge, identify relevant variables, and refine data requirements by distinguishing causal relationships from mere correlations before collecting the training data. Particularly, graphical causal models in the form of directed acyclic graph (DAG) allow to communicate explicitly assumed directions of causality and assumptions about *confounders*, i.e., situations in which a variable Z is associated to two random variables X_1 and X_2 such that a *spurious relationship* between X_1 and X_2 can be observed: $X_1 \leftarrow Z \rightarrow X_2$.

Figure 1 outlines a proposed workflow. The workflow bases on the principle of *causal factorisation* [25]:

$$p(X_1, \dots, X_n) = \prod_{i=1}^n p(X_i | \mathbf{PA}_i), \quad (1)$$

where \mathbf{PA}_i denotes the set of parents (variables that have a direct causal effect) of a variable X_i in the DAG. *Causal factorisation* implies that an observed joint distribution of interest can be decomposed into a product of conditional distributions, where each term corresponds to a causal mechanism.

Step 1) Identify individual causal mechanisms: The aim is to pinpoint specific cause-effect pathways informed based on high-level requirements, prior domain knowledge, and context assumptions.

Step 2) Update causal graph: Once a causal mechanism is identified, the relevant observable and latent variables are determined, and a causal graph is updated to include these variables along with the assumed directions of cause-and-effect relationships.

Step 3) Perform d-separation and extract requirements:

With the causal model, *d-separation*¹ allows to identify variables that are needed to block “non-causal” association paths. Taking the example from above, in $X_1 \leftarrow Z \rightarrow X_2$, there is a “non-causal” path between X_1 and X_2 . If the ML model can condition on Z (assuming Z is observable), X_1 and X_2 become d-separated, closing the “non-causal” path. This is an example of a resulting data requirement: Z must be included in the training dataset to avoid learning a spurious correlation between X_1 and X_2 . Additionally, Step 3 provides *independence criteria* based on global Markov properties: If X_1 and X_2 are d-separated by Z , they are conditionally independent given Z , i.e., $X_1 \perp X_2 | Z$. This provides *testable criteria* to verify prior knowledge and assumptions encoded in the causal graph.

Step 4) Check consistency and observability: The graphical causal model must be checked for cyclic dependencies because a variable cannot be its own cause [13]. Furthermore, variables needed to block “non-causal” paths must be observable. If this is not the case, the system must be adjusted to enable their observation or suitable instrument variables must be identified [3].

The resulting causal graph becomes part of an ML specification because it encodes the assumed causal structures, prior knowledge, and operational context, from which data and model requirements, as well as testing criteria, are derived.

4 Demonstration on industrial prototype

We demonstrate the use of causal modelling as an RE activity on an industrial prototype use case, specifically a system for detecting faults in the cooling system of electric motors.

Methodology: We held three workshops with two Siemens engineers and two academic researchers to explore using causal models for requirements specification in the second half of 2022. The researchers introduced causal models with examples like *temperature* \leftarrow *sunrise* \rightarrow *birds chirping* and explaining key concepts such as *confounding*, *colliders*, and *d-separation* using for example the *back-door criteria*. The company experts then presented the prototype system, and prior knowledge rules were formalised together by identifying causal mechanisms and updating the causal model iteratively with each newly found causal mechanism. We then applied *d-separation* to close non-causal paths between the exposure (i.e., a cooling fault) and the outcome (i.e., the classification result) which resulted in data and model requirements to ensure the ML model controls for potential confounding.

Description of demonstration case: The demonstration case, provided by Siemens, is a motor diagnostic application for monitoring electrical motors using an attachable sensor device. Initially, the system detected cooling faults from vibrations caused by mechanical faults, such as a broken fan blade. The new device will use an ML model to detect faults based on multiple sensor inputs. The high-level functional requirement is:

FR-1: *GIVEN indoor operational environment WHEN the cooling system is abnormal THEN an alarm should be raised.*

The following prior knowledge of the company engineers was considered for identifying causal mechanisms:

¹Due to space constraints, background on d-separation is omitted but can be found in [22].

Table 1: Variables for motor diagnostic use case.

Variable	Related PKs	Definition
Cooling Fault	PK1, PK2	Fan system status
Q	PK2	Max. possible airflow
Mech. Fault	PK2, PK3	Mechanical fault of motor
P_M	PK1	Mechanical power
R_1	PK1	Electrical (inner) losses
T_E	PK4	Environmental temperature
U_X	PK5	Unmeasured noises
$T(T_s)$		Surface temperature (measured)
$H(H_s)$		Magnetic Flux (measured)
$V(V_s)$		Vibrations (measured)

PK-1: A fault in the cooling system can affect the magnetic flux by changing the temperature of the rotor material and thus affecting the electrical resistance.

PK-2: Mechanical faults of the fan can reduce the available airflow.

PK-3: Mechanical faults cause vibrations of the system.

PK-4: Environmental temperature has an influence on the temperature signal because the sensor is mounted outside the motor.

PK-5: Unmeasured sensor disturbances exist in general.

Results: The resulting causal model for the motor diagnostic use case is shown in Figure 2.

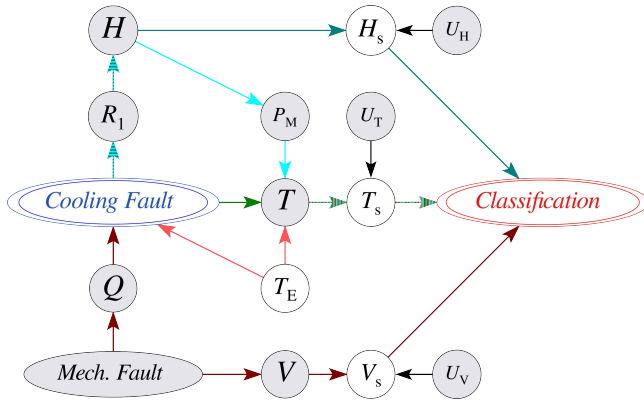


Figure 2: DAG for the motor diagnostic use case. Gray-background nodes are latent (unobservable) variables, while white-background nodes are observable at runtime.

Explanations for the variables and their relations to the prior knowledge are provided in Table 1.

The causal graph in Figure 2 includes three causal mechanisms between *Cooling Fault* occurrence and *Classification* whether or not a cooling fault has occurred:

Temperature mechanism: A cooling fault increases the motor's surface temperature T (via the core temperature), measured by the temperature sensor T_s , which can be used to classify a cooling fault.

Magnetic flux mechanism: A cooling fault changes the inner resistance (via the core temperature), which affect the magnetic flux H . This is measured by the fluxmeter H_s for classification.

Table 2: Requirements derived from causal graph

ID	Requirement (RQ-D: Data Req., RQ-M: Model Req.)
RQ-D1	Training data shall include cases where mechanical faults cause vibrations V without leading to cooling faults.
RQ-D2	The occurrence of cooling faults shall be conditioned on different environmental temperatures T_E such that the model can learn the confounding influence of T_E .
RQ-D3	Measurements shall include characteristic sensor noise.
RQ-M1	Cooling faults shall not be classified based on vibration data V_s alone.
RQ-M2	The input layer shall accept temperature, magnetic flux, and vibration measurements.

Mechanical power mechanism: A cooling fault changes the magnetic flux H , which affects the mechanical power P_M and surface temperature T . The latter is measured by the sensor T_s for classification.

Two confounding paths were identified:

Mechanical fault confounding: A mechanical fan blade fault can reduce the available airflow Q causing a cooling fault and vibrations V , which are measured by a vibration sensor V_s for classification.

Environmental temperature confounding: A sudden change in environment temperature T_E can temporarily limit cooling without indicating a fault and it affects the surface temperature T .

Data and model requirements: We checked which variables must be observed and controlled for to close non-causal paths between cooling fault occurrence and classification of a cooling fault, which resulted in the requirements listed in Table 2. Vibration data V_s alone is insufficient to detect cooling faults, as not all mechanical faults lead to a cooling fault (RQ-D1, RQ-M1). Instead, data on temperature and magnetic flux mechanisms should be included (RQ-M2).² An additional sensor should record the environmental temperature T_E to control for confounding (RQ-D2). Sensor noise must also be represented in the training data (RQ-D3).

Testing and runtime checks: The causal graph in Figure 2 implies a set of independence conditions:

ID1 : $\text{Classification} \perp T_E \mid H_s, T_s, V_s$

ID2 : $H_s \perp T_E \mid \text{Cooling Fault}$

ID3 : $H_s \perp V_s \mid \text{Cooling Fault}$

ID4 : $T_s \perp V_s \mid \text{Cooling Fault}, T_E$

ID5 : $V_s \perp T_E$

As an example for a resulting test case, ID1 states that classification is independent of T_E given H_s , T_s , and V_s . A test case could trigger faults at varying T_E to verify that the detection probability remains unchanged. As an example for runtime monitoring, ID5 suggests V_s and T_E should be independent. An additional monitor could track their correlation during operation and trigger an alarm if a threshold is exceeded which would indicate a shift in the assumed operational context (e.g., the probability of a mechanical fault could depend on

²In fact, vibration data may be unnecessary for detecting cooling faults unless it is desired to distinguish mechanical from non-mechanical causes.

the environmental temperature which would be a violation of the assumed causal models for this system).

5 Discussion and research agenda

In this vision paper we argue that causal modelling and its mathematical framework have significant potential as an RE activity for ML software system development by systematically integrating prior knowledge into the design. However, based on the experience in our demonstration use case, further research is needed before this vision becomes standard industry practice.

Causal models as complement to natural language requirements. Causal graphs originates from mathematics. We must explore how they can complement current requirements specifications and how they must be adopted for RE. Terms like “treatment”, “confounder”, and “collider” are uncommon in RE and require interpretation.

Criteria for sufficient variable selection. A key challenge is knowing when a causal graphs includes “enough” prior knowledge. We need methods to determine a sufficient set of variables that must be included for a given use case and methods for deciding between competing causal DAGs given a “sufficient” set of variables.

Modularisation of ML software systems. Isolating causal mechanisms can guide the modularisation of ML systems, i.e., dividing large monolithic ML models into smaller sub-models.

A common language between different stakeholders of ML software systems. Causal models provide a unified way to communicate prior knowledge and assumptions. Research should explore how this can facilitate coordination between different groups such as data scientists, product experts, and software engineers.

Data requirements derived through causal reasoning. Causal reasoning in RE helps identifying data requirements. Further research should assess to what degree data requirements derived from causal models can enhance ML robustness and reduce data needs compared to traditional RE methods.

Testing and runtime checks. ML software system must align with expected (causal) behaviour. Causal graphs imply independence criteria that lead to *testable implications* for the runtime behaviour. Research should explore how to translate these into testing strategies and monitors and how reliable such monitors are in practice.

Conclusion. Causal reasoning offers a systematic way to integrate prior knowledge into RE for ML software systems. We outlined a vision and demonstrated a preliminary workflow to derive and argue for low level model and data requirements from high level prior knowledge using causal graphs. We discussed future research activities that are needed to turn this vision into industrial practice.

Acknowledgments

This project has received funding from the EU’s Horizon research and innovation program under grant agreement No 957197 (vedliot).

References

- [1] Khlood Ahmad, Mohamed Abdelrazek, Chetan Arora, Muneera Bano, and John Grundy. 2023. Requirements engineering for artificial intelligence systems: A systematic mapping study. *IST* 158 (2023), 107176.
- [2] Antonio Pedro Santos Alves, Marcos Kalinowski, Giray, et al. 2023. Status quo and problems of requirements engineering for machine learning: Results from an international survey. In *Int. Conf. on Product-Focused Soft. Proc. Improv.* Springer, 159–174.
- [3] Joshua D Angrist, Guido W Imbens, and Donald B Rubin. 1996. Identification of causal effects using instrumental variables. *Journal of the American Stat. Asso.* 91, 434 (1996), 444–455.
- [4] Markus Borg, Cristofer Englund, Wnuk, et al. 2019. Safely Entering the Deep: A Review of Verification and Validation for Machine Learning and a Challenge Elicitation in the Automotive Industry. *Journal of Automotive Software Engineering* 1, 1 (2019), 1–19.
- [5] Alexander D’Amour, Katherine Heller, Moldovan, et al. 2022. Underspecification presents challenges for credibility in modern machine learning. *Journal of Machine Learning Research* 23, 226 (2022), 1–61.
- [6] Alessandro Fantechi, Alessio Ferrari, Stefania Gnesi, and Laura Semini. 2018. Requirement engineering of software product lines: Extracting variability using NLP. In *2018 26th IEEE RE Conf. IEEE*, 418–423.
- [7] Jannik Fischbach, Julian Frattini, Arjen Spaans, Maximilian Kummeth, Andreas Vogelsang, Daniel Mendez, and Michael Unterkalmsteiner. 2021. Automatic detection of causality in requirement artifacts: the cira approach. In *2021 27th REFSQ*. Springer, 19–36.
- [8] Jannik Fischbach, Julian Frattini, Andreas Vogelsang, et al. 2023. Automatic creation of acceptance tests by extracting conditionals from requirements: NLP approach and case study. *JSS* 197 (2023), 111549.
- [9] Jannik Fischbach, Benedikt Hauptmann, Lukas Konwitschny, Dominik Spies, and Andreas Vogelsang. 2020. Towards causality extraction from requirements. In *2020 28th IEEE RE Conf. IEEE*, 388–393.
- [10] Xavier Franch, Andreas Jedlitschka, and Silverio Martínez-Fernández. 2023. A requirements engineering perspective to AI-based systems development: A vision paper. In *2023 REFSQ*. Springer, 223–232.
- [11] Julian Frattini, Davide Fucci, Richard Torkar, Lloyd Montgomery, Michael Unterkalmsteiner, Jannik Fischbach, and Daniel Mendez. 2025. Applying bayesian data analysis for causal inference about requirements quality: a controlled experiment. *Empirical Software Engineering* 30, 1 (2025), 29.
- [12] Luca Giamattei, Antonio Guerriero, Roberto Pietrantuono, and Stefano Russo. 2024. Causal reasoning in Software Quality Assurance: A systematic review. *IST* (2024), 107599.
- [13] Madelyn Glymour, Judea Pearl, and Nicholas P Jewell. 2016. *Causal inference in statistics: A primer*. John Wiley & Sons.
- [14] Lucas Gren and Richard Berntsson Svensson. 2021. Is it possible to disregard obsolete requirements? a family of experiments in software effort estimation. *REJ* 26, 3 (2021), 459–480.
- [15] Noah Jadallah, Jannik Fischbach, Julian Frattini, and Andreas Vogelsang. 2021. Cate: Causality tree extractor from natural language requirements. In *2021 29th RE Conf. Workshops (REW)*. IEEE, 77–79.
- [16] Alessia Knauss, Daniela Damian, Xavier Franch, Angela Rook, Hausi A Müller, and Alex Thomo. 2016. ACon: A learning-based approach to deal with uncertainty in contextual requirements at runtime. *IST* 70 (2016), 85–99.
- [17] Alessia Knauss, Daniela Damian, and Kurt Schneider. 2014. Eliciting contextual requirements at design time: A case study. In *2014 4th Int. Workshop on Emp. RE. IEEE*, 56–63.
- [18] Hiroshi Kuwajima, Hirotoshi Yasuoka, and Toshihiro Nakae. 2020. Engineering problems in machine learning systems. *MLg* 109, 5 (2020), 1103–1126.
- [19] Robert Maier, Lisa Grabinger, David Uhlhart, and Jürgen Mottok. 2022. Towards causal model-based engineering in automotive system safety. In *Int. Symp. on Model-Based Safety*. Springer, 116–129.
- [20] Robert Maier, Andreas Schlattl, Thomas Guess, and Jürgen Mottok. 2024. CausalOps—Towards an industrial lifecycle for causal probabilistic graphical models. *IST* 174 (2024), 107520.
- [21] Shira Mitchell, Eric Potash, Solon Barocas, Alexander D’Amour, and Kristian Lum. 2021. Algorithmic fairness: Choices, assumptions, and definitions. *Ann. Review of Stat. and Its Appl.* 8 (2021), 141–163.
- [22] Judea Pearl. 2019. The Limitations of Opaque Learning Machines. In *Possible Minds: 25 Ways of Looking at AI*, Johnm Brockman (Ed.). Penguin Press, London, Chapter 2.
- [23] Zhongyi Pei, Lin Liu, Chen Wang, and Jianmin Wang. 2022. Requirements engineering for machine learning: A review and reflection. In *2022 30th IEEE RE Conf. Workshops (REW)*. IEEE, 166–175.
- [24] Mona Rahimi, Jin LC Guo, Sahar Kokaly, and Marsha Chechik. 2019. Toward requirements specification for machine-learned components. In *2019 27th IEEE RE Workshops (REW)*. IEEE, 241–244.
- [25] Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. 2021. Toward causal representation learning. *Proc. IEEE* 109, 5 (2021), 612–634.
- [26] Hugo Villamizar, Marcos Kalinowski, Hélio Lopes, and Daniel Mendez. 2024. Identifying concerns when specifying machine learning-enabled systems: a perspective-based approach. *JSS* 213 (2024), 112053.